Tle

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/538,132	03/29/2000	Heng Liao	16491-002710US	6523

20350 7590 02/23/2006

TOWNSEND AND TOWNSEND AND CREW, LLP
TWO EMBARCADERO CENTER
EIGHTH FLOOR
SAN FRANCISCO, CA 94111-3834

EXAMINER	
EL CHANTI, HUSSEIN A	

ART UNIT	PAPER NUMBER
2157	

DATE MAILED: 02/23/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/538,132	LIAO, HENG	
	Examiner	Art Unit	
	Hussein A. El-chanti	2157	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *01 December 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-54* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-54* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

Response to Amendment

1. This action is responsive to amendment received on Dec. 1, 2005. Claims 1-54 are pending examination.

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

2. Claims 1-4, 11-12, 18-22, 47 and 50 are rejected under 35 U.S.C. 102(b) as being anticipated by Hekhuis, U.S. Patent No. 5,414,650.

Hekhuis teaches the invention explicitly as claimed including a system and method for parsing incoming packets and classifying the packet flow (see abstract)

As to claim 1, Hekhuis teaches a method for classifying data packets comprising the steps of:

Providing a language definition (see col. 8 lines 55-col. 9 lines 6, a hash table of words are provided to provide language deifinition); and

Processing incoming network data with said language definition in accordance with a formal language processing technique including scanning said network data using lexical token scanning according to said language definition, wherein said network data is treated as a stream of input bytes, said network data being organized into data packets, said scanning resulting in the identification of a data packet as belonging to one of a plurality of classes (see col. 8 lines 55-col. 9 lines 37 and col.

10 lines 40-col. 11 lines 42, packets are parsed according to parsing rules to identify words where packets are classified accordingly).

As to claim 2, Hekhuis teaches the method of claim 1 wherein said scanning includes identifying an arithmetic operation and performing said arithmetic operation (see col. 8 lines 55-col. 9 lines 37 and col. 10 lines 40-col. 11 lines 42).

As to claims 3, Hekhuis teaches the method of claim 1 wherein said scanning includes identifying a skip operation and in response thereto skipping over one or more subsequent input bytes (see col. 8 lines 55-col. 9 lines 37 and col. 10 lines 40-col. 11 lines 42).

As to claims 4, Hekhuis teaches the method of claim 1 wherein said lexical scanning includes providing a set of regular expressions, each regular expression having an associated class identifier (see col. 8 lines 55-col. 9 lines 37 and col. 10 lines 40-col. 11 lines 42).

As to claim 11, Hekhuis teaches a method for classifying data packets comprising steps of: providing a language definition in the form of one or more regular expressions, each having an associated class identifier; receiving plural data packets, each having a length not necessarily equal to one another; and for each data packet, processing it in accordance with a formal language processing technique including determining a matching regular expression from among said regular expressions that matches said data packet, wherein said each data packet is classified according to the class identifier associated with said matching regular expression (see col. 8 lines 55-col. 9 lines 37 and col. 10 lines 40-col. 11 lines 42).

As to claim 12, Hekhuis teaches the method of claim 11 wherein said data packets comprise a data stream and said determining includes lexically scanning said data stream (see col. 8 lines 55-col. 9 lines 37 and col. 10 lines 40-col. 11 lines 42).

As to claim 18, Hekhuis teaches the method of claim 11 wherein said data packet comprises plural bytes, and said determining includes detecting an operator indicating a number of bytes to be skipped (see col. 8 lines 55-col. 9 lines 37 and col. 10 lines 40-col. 11 lines 42).

As to claim 19, Hekhuis teaches the method of claim 18 wherein said number is specified by the value of a current input byte (see col. 8 lines 55-col. 9 lines 37 and col. 10 lines 40-col. 11 lines 42).

As to claim 20, Hekhuis teaches the method of claim 18 wherein said number is specified in a register (see col. 8 lines 55-col. 9 lines 37 and col. 10 lines 40-col. 11 lines 42).

As to claim 21, Hekhuis teaches the method of claim 18 wherein said determining further includes detecting an operator indicating a value to be saved in a register (see col. 8 lines 55-col. 9 lines 37 and col. 10 lines 40-col. 11 lines 42).

As to claim 22, Hekhuis teaches the method of claim 21 wherein said determining further includes detecting an operator indicating a logical or mathematical operation to be performed on the contents of said register (see col. 8 lines 55-col. 9 lines 37 and col. 10 lines 40-col. 11 lines 42).

As to claim 47, Hekhuis teaches a network packet classifier comprising: means for receiving an incoming network packet; means for processing said network packet in

accordance using a language definition in accordance with a formal language ,processing technique including classifying said network packet by matching the pattern of its constituent data against plural regular expressions, each regular expression having a corresponding class identifier; and means for outputting a class identifier of the regular expression which matches said network packet (see col. 8 lines 55-col. 9 lines 37 and col. 10 lines 40-col. 11 lines 42).

As to claim 50, Hekhuis teaches the classifier of claim 47 wherein some of said states have one or more associated instructions, the classifier further including an arithmetic logic unit operatively coupled to said system of logic circuits and configured to execute said instructions (see col. 8 lines 55-col. 9 lines 37 and col. 10 lines 40-col. 11 lines 42).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 5-10, 13-17, 23-45, 48-49 and 51-54 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hekhuis in view of Narad.

As to claim 5, Hekhuis teaches a method for classifying data packets comprising processing incoming network data with said language definition in accordance with a formal language processing technique including scanning said network data using lexical token scanning according to said language definition, wherein said network data

is treated as a stream of input bytes, said network data being organized into data packets, said scanning resulting in the identification of a data packet as belonging to one of a plurality of classes (see col. 8 lines 55-col. 9 lines 37 and col. 10 lines 40-col. 11 lines 42).

Hekhuis does not explicitly teach the "providing a deterministic finite automaton DFA comprising plural states, said step of scanning including recognizing data packets using said DFA including transitioning from one state to another".

However Narad teaches a method of parsing packets and generating DFA to identify a packet classification level (see col. 36 lines 19-col. 37 lines 22 and col. 103-col. 104).

It would have been obvious for one of the ordinary skill in the art at the time of the invention to modify Hekhuis in view of including a DFA as taught by Narad. One would be motivated to include a DFA including recognizing the lexical tokens using the DFA in Riddle because doing so would allow the DFA to determine the classification of the data packet and map an ordered sequence of input events into a corresponding sequence according to the control section of the data where the next state is uniquely determined by a single input event.

As to claim 6, Hekhuis teaches the method of claim 5 wherein said data packets are variable length data packets (see col. 8 lines 55-col. 9 lines 37 and col. 10 lines 40-col. 11 lines 42).

As to claim 7, Narad teaches the method of claim 5 wherein said DFA is defined by a set of regular expressions (see col. 36 lines 19-col. 37 lines 22 and col. 103-col. 104).

As to claim 8, Hekhuis does not explicitly teach regular expression, producing a non-deterministic finite automaton (NFA) from said grammar tree, data structure, and converting said NFA to produce said DFA".

However Narad teaches a system and method for parsing data packets where to generate NFA (see col. 36 lines 45-60 and col. 103-col. 104).

It would have been obvious for one of the ordinary skill in the art at the time of the invention to modify Hekhuis by producing a NFA and converting NFA to produce DFA because doing so would provide faster processing since processing of DFA takes less time than processing NFA.

As to claim 9, Hekhuis teaches the method of claim 5 wherein some of said states include one or more associated computer instructions and wherein said computer instructions are executed in connection with transitioning to a state (see col. 8).

As to claim 10, Hekhuis teaches the method of claim 9 wherein some of said states further include a skip instruction (see col. 9-col. 10).

As to claim 13, Narad teaches regular expressions are represented by a deterministic finite automaton (DFA) (see col. 103-col. 104).

As to claim 14, Narad teaches DFA is in compressed form (see col. 8 lines 65-col. 9 lines 30 and col. 36 lines 19-col. 37 lines 22).

As to claim 15, Narad teaches compiling said regular expressions to produce said DFA (see col. 8 lines 65-col. 9 lines 30 and col. 36 lines 19-col. 37 lines 22 and col. 103-col. 104).

As to claim 16, Narad teaches said compiling produces a non-deterministic finite automaton (NFA) as intermediate data structure, said compiling further includes converting said NFA to produce said DFA (see col. 8 lines 65-col. 9 lines 30 and col. 36 lines 19-col. 37 lines 22 and col. 103-col. 104).

As to claim 17, Narad teaches reducing said DFA to a compressed form (see col. 8 lines 65-col. 9 lines 30 and col. 36 lines 19-col. 37 lines 22 and col. 103-col. 104).

As to claim 23, Hekhuis teaches a method for classifying data packets comprising processing incoming network data with said language definition in accordance with a formal language processing technique including scanning said network data using lexical token scanning according to said language definition, wherein said network data is treated as a stream of input bytes, said network data being organized into data packets, said scanning resulting in the identification of a data packet as belonging to one of a plurality of classes (see col. 8 lines 55-col. 9 lines 37 and col. 10 lines 40-col. 11 lines 42).

Hekhuis does not explicitly teach the "providing a deterministic finite automaton DFA comprising plural states, said step of scanning including recognizing data packets using said DFA including transitioning from one state to another".

However Narad teaches a method of parsing packets and generating DFA to identify a packet classification level (see col. 36 lines 19-col. 37 lines 22 and col. 103-col. 104).

It would have been obvious for one of the ordinary skill in the art at the time of the invention to modify Hekhuis in view of including a DFA as taught by Narad. One would be motivated to include a DFA including recognizing the lexical tokens using the DFA in Riddle because doing so would allow the DFA to determine the classification of the data packet and map an ordered sequence of input events into a corresponding sequence according to the control section of the data where the next state is uniquely determined by a single input event.

As to claims 24, Hekhuis teaches the method of claim 23 wherein said scanning includes identifying an arithmetic operation and performing said arithmetic operation (see col. 8 lines 55-col. 9 lines 37 and col. 10 lines 40-col. 11 lines 42).

As to claim 25, Hekhuis teaches the method of claim 23 wherein said lexical scanning includes providing a set of regular expressions, each regular expression having an associated class identifier (see col. 8 lines 55-col. 9 lines 37 and col. 10 lines 40-col. 11 lines 42).

As to claim 26, Hekhuis teaches the method of claim 23 wherein said determining further includes detecting an operator indicating a logical or mathematical operation to be performed on the contents of said register (see col. 8 lines 55-col. 9 lines 37 and col. 10 lines 40-col. 11 lines 42).

As to claim 27, Hekhuis teaches the method of claim 26 wherein said scanning includes identifying a skip operation and in response thereto skipping over one or more subsequent input bytes (see col. 8 lines 55-col. 9 lines 37 and col. 10 lines 40-col. 11 lines 42).

As to claim 28, Hekhuis teaches the method of claim 27 wherein said regular expressions further include data storage operations(see col. 8 lines 55-col. 9 lines 37 and col. 10 lines 40-col. 11 lines 42).

As to claim 29, Hekhuis teaches the method of claim 23 wherein said DFA is in compressed form (see col. 8 lines 65-col. 9 lines 30 and col. 36 lines 19-col. 37 lines 22).

As to claim 30, Hekhuis teaches method of claim 23 comprising:

Receiving a description of grammar rules in the form of a grammar packet classification language;

Compiling said grammar packet classification language to produce a grammar graph;

Configuring a classifier with said grammar graph;

Processing said data stream in accordance with a formal language processing technique using said grammar packet classification language including parsing said data stream with said grammatical packet classifier to identify a protocol structure in a received data packet; and

Processing said received data packet in accordance with said protocol structure (see col. 8 lines 55-col. 9 lines 37 and col. 10 lines 40-col. 11 lines 42).

As to claim 31, Hekhuis teaches a method for classifying data packets comprising processing incoming network data with said language definition in accordance with a formal language processing technique including scanning said network data using lexical token scanning according to said language definition, wherein said network data is treated as a stream of input bytes, said network data being organized into data packets, said scanning resulting in the identification of a data packet as belonging to one of a plurality of classes (see col. 8 lines 55-col. 9 lines 37 and col. 10 lines 40-col. 11 lines 42).

Hekhuis does not explicitly teach the "providing a deterministic finite automaton DFA comprising plural states, said step of scanning including recognizing data packets using said DFA including transitioning from one state to another".

However Narad teaches a method of parsing packets and generating DFA to identify a packet classification level (see col. 36 lines 19-col. 37 lines 22 and col. 103-col. 104).

It would have been obvious for one of the ordinary skill in the art at the time of the invention to modify Hekhuis in view of including a DFA as taught by Narad. One would be motivated to include a DFA including recognizing the lexical tokens using the DFA in Riddle because doing so would allow the DFA to determine the classification of the data packet and map an ordered sequence of input events into a corresponding sequence according to the control section of the data where the next state is uniquely determined by a single input event.

As to claim 32, Narad teaches the classifier some of said regular expressions include arithmetic instructions and logic instructions, said memory assemblage further configured to contain said instructions, the classifier further including an arithmetic logic unit operatively coupled to said decompression logic and configured to execute said instructions (see col. 4 lines 47-67, col. 9 lines 1-35, col. 8 lines 50-65 and col. 36 lines 19-col. 37 lines 22).

As to claim 33, Narad teaches the classifier of claim 32 further including at least one register operatively coupled to said arithmetic logic unit, said arithmetic logic unit further configured to store data into said register in response to a save instruction (see col. 8 lines 65-col. 9 lines 30 and col. 36 lines 19-col. 37 lines 22).

As to claim 34, Hekhuis teaches the classifier of claim 32 further including skip logic operatively coupled to said logic component and configured to skip over an amount of data in response a skip instruction (see col. 8).

As to claim 35, Hekhuis teaches the classifier of claim 31 wherein said network data packets can vary from one packet to another (see col. 10).

As to claim 36, Narad teaches the classifier of claim 31 wherein said DFA is in compressed form (see col. 8 lines 65-col. 9 lines 30 and col. 36 lines 19-col. 37 lines 22).

As to claim 37, Narad teaches the classifier of claim 36 wherein said DFA comprises plural non-default states and plural default states, and said memory assemblage comprises a base memory, a next-state memory, and a default-state memory; said base memory configured to contain address locations of said next-state

memory, said next-state memory representing all of said non-default states, said default-state memory representing all of said default states (see col. 8 lines 65-col. 9 lines 30 and col. 36 lines 19-col. 37 lines 22).

As to claim 38, Narad teaches the classifier of claim 37 wherein said memories are random access memories (see col. 42 lines 35-67).

As to claim 39, Narad teaches the classifier of claim 37 wherein said memories are read only memories (see col. 42 lines 35-67).

Hekhuis teaches a method for classifying data packets comprising processing incoming network data with said language definition in accordance with a formal language processing technique including scanning said network data using lexical token scanning according to said language definition, wherein said network data is treated as a stream of input bytes, said network data being organized into data packets, said scanning resulting in the identification of a data packet as belonging to one of a plurality of classes (see col. 8 lines 55-col. 9 lines 37 and col. 10 lines 40-col. 11 lines 42).

Hekhuis does not explicitly teach the "providing a deterministic finite automaton DFA comprising plural states, said step of scanning including recognizing data packets using said DFA including transitioning from one state to another".

However Narad teaches a method of parsing packets and generating DFA to identify a packet classification level (see col. 36 lines 19-col. 37 lines 22 and col. 103-col. 104).

It would have been obvious for one of the ordinary skill in the art at the time of the invention to modify Hekhuis in view of including a DFA as taught by Narad. One

would be motivated to include a DFA including recognizing the lexical tokens using the DFA in Riddle because doing so would allow the DFA to determine the classification of the data packet and map an ordered sequence of input events into a corresponding sequence according to the control section of the data where the next state is uniquely determined by a single input event.

As to claim 41, Narad teaches the classifier of claim 40 further including a third system of memory configured to contain current state information for plural input channels, said system of logic circuits operatively coupled to said third system of memory to initialize said DFA in accordance with current state information corresponding to the input channel associated with said data packet.

As to claim 42, Hekhuis teaches the classifier of claim 40 wherein some of said states have one or more associated instructions, the classifier further including an arithmetic logic unit operatively coupled to said system of logic circuits and configured to execute said instructions (see col. 8 lines 65-col. 9 lines 30 and col. 36 lines 19-col. 37 lines 22).

As to claim 43, Hekhuis teaches the classifier of claim 42 further including at least one register operatively coupled to said arithmetic logic unit, said arithmetic logic unit further configured to store data into said register in response to a save instruction (see col. 8 lines 65-col. 9 lines 30 and col. 36 lines 19-col. 37 lines 22).

As to claim 44, Hekhuis teaches the classifier of claim 42 further including skip logic operatively coupled to said logic component and configured to skip over an amount of data in response a skip instruction (see col. 8-col. 10).

As to claim 45, Hekhuis teaches the classifier of claim 40 wherein said stream of data is a stream of bytes (see col. 8 lines 65-col. 9 lines 30 and col. 36 lines 19-col. 37 lines 22).

As to claim 46, Hekhuis teaches the classifier of claim 40 wherein said network data packets can vary from one packet to another (see col. 8-10).

As to claim 48, Narad teaches scanning includes a memory component configured with data to represent a deterministic finite automaton (DFA) (see col. 8 lines 65-col. 9 lines 30 and col. 36 lines 19-col. 37 lines 22).

As to claim 49, Hekhuis teaches the classifier of claim 48 wherein said means for outputting includes a second memory component configured with said class identifiers (see col. 8 lines 65-col. 9 lines 30 and col. 36 lines 19-col. 37 lines 22).

As to claim 51, Hekhuis teaches a method for classifying data packets comprising processing incoming network data with said language definition in accordance with a formal language processing technique including scanning said network data using lexical token scanning according to said language definition, wherein said network data is treated as a stream of input bytes, said network data being organized into data packets, said scanning resulting in the identification of a data packet as belonging to one of a plurality of classes (see col. 8 lines 55-col. 9 lines 37 and col. 10 lines 40-col. 11 lines 42).

Hekhuis does not explicitly teach the "providing a deterministic finite automaton DFA comprising plural states, said step of scanning including recognizing data packets using said DFA including transitioning from one state to another".

However Narad teaches a method of parsing packets and generating DFA to identify a packet classification level (see col. 36 lines 19-col. 37 lines 22 and col. 103-col. 104).

It would have been obvious for one of the ordinary skill in the art at the time of the invention to modify Hekhuis in view of including a DFA as taught by Narad. One would be motivated to include a DFA including recognizing the lexical tokens using the DFA in Riddle because doing so would allow the DFA to determine the classification of the data packet and map an ordered sequence of input events into a corresponding sequence according to the control section of the data where the next state is uniquely determined by a single input event.

As to claim 52, Hekhuis teaches the classifier of claim 51 wherein said network data packets can vary from one packet to another (see col. 8 lines 65-col. 9 lines 30 and col. 36 lines 19-col. 37 lines 22).

As to claim 53, Hekhuis teaches the classifier of claim 51 wherein said regular expressions include arithmetic and logic operations (see col. 8 lines 65-col. 9 lines 30 and col. 36 lines 19-col. 37 lines 22).

As to claim 54, Hekhuis teaches the classifier of claim 51 wherein said regular expressions further include skip operations (see col. 8 lines 65-col. 9 lines 30 and col. 36 lines 19-col. 37 lines 22).

Response to Arguments

4. Applicant's arguments have been fully considered but are not persuasive. Applicant argues in substance that Hekhuis does not disclose parsing packets in accordance with a formal language.

In response, Hekhuis teaches a system and method for parsing packets and classifying packets according to rules (see abstract). The system parses packets according to parsing rules. For example, the system may identify a packet as a cardinal packet by comparing the string identified from the packet to the set {"and", "br", "d", "f", "ing", "th"}. The matching of the string with the set {"and", "br", "d", "f", "ing", "th"} to identify a cardinal packet is interpreted by examiner to be the grammar used to parse the packets (see col. 8 lines 55-col. 9 lines 6). In addition, the received packets are compared to a dictionary i.e. "formal language", if the packet information is not in the dictionary, a token is assigned to the packet (see col. 12 lines 52-col. 13 lines 5). Therefore Hekhuis's comparing of packet information to a dictionary and adding tokens to the packets in accordance with the comparison meets the scope of the claimed limitation "processing incoming network data in accordance with a formal language".

5. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any

extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Hussein A. El-chanti whose telephone number is (571)272-3999. The examiner can normally be reached on Mon-Fri 8:30-5:00.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ario Etienne can be reached on (571)272-4001. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Hussein El-chanti

Feb. 5, 2006

ARIO ETIENNE
PRIMARY EXAMINER